Beyond Go-Live: Protecting the Value of Your Technology Investment

Spark Healthcare Advisory · Oct 22, 2026

SPARK HEALTHCARE ADVISORY · INSIGHTS

Launch day is one day.
The return is decided in the months around it.

Readiness · Go-live · The first 90 days · Retirement · Measuring value

Spark Healthcare Advisory · Insights

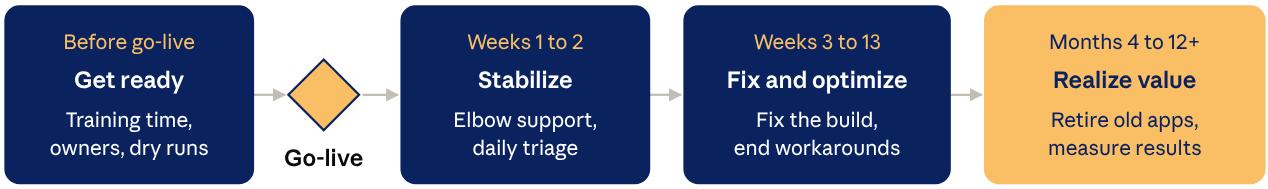
Most technology projects are planned around one date: go-live. The budget, the timeline, and the vendor's staffing all point at that day. But the value of the investment depends on what happens around it: whether people were ready going in, and whether the system keeps getting better after the project team leaves.

Launch day gets the attention. The weeks before it and the months after it decide whether the investment pays off.

Below are the five places we see value won or lost, and the questions leaders can ask at each one.

Most of the value is won before and after launch day

Where the work sits in a typical implementation



Often left out of the plan

implementation phases · 4 phases, 1 gate

Launch day is a single gate. The work that decides the return happens in the phases on either side of it.

1. Readiness is a capacity problem

Most readiness gaps are not knowledge gaps. They are time gaps. The plan is signed and the timeline looks good, but the people who need to get ready still have full-time jobs.

Training gets squeezed between shifts. Decisions wait on someone who is already stretched thin. A new workflow looks clear on paper, but the people doing the work still have questions nobody has had time to answer.

What helps:

- **Protect training time like a clinical shift.** Put it on the schedule and fund backfill, so people are not choosing between training and patient care.
- **Name who decides.** For each area, one person who can make a call when something stalls, with a clear path above them.
- **Walk through a real day.** Have staff run the work they will actually do, start to finish, using their own scenarios. Dry runs catch what slide decks miss.
- **Spot the overloaded people early.** The same few experts often end up as trainers, testers, and super users at once. Spread the load or bring in help.
- **Make readiness visible.** Track training completion, open build issues, and unanswered workflow questions by department, so leaders see the risk before the date arrives.

If you only do one thing, ask each department leader: "Does your team actually have the time to get ready?" Then believe the answer.

A readiness scorecard

Readiness is easier to manage when every department is measured the same way. A simple scorecard, reviewed weekly starting about 90 days out, turns a general feeling into a list of specific gaps.

Area	What ready looks like	Red flag
Training	Every user has finished role-based training and practiced their own workflows	One unit or shift lags behind, often nights and weekends
Build and testing	Critical workflows are tested end to end with real scenarios, and every open defect has an owner and a date	High-severity defects are still open in the final 30 days

Area	What ready looks like	Red flag
Decisions	Every open workflow question has an answer and an owner	The same questions keep coming back to steering meetings
Staffing	Schedules include backfill for training and go-live support	Overtime is already high and the plan is to push through
Devices and access	Every user can log in, and every printer, scanner, and badge reader works	Hardware and login testing is scheduled for the final week
Communication	Staff know the date, what changes for them, and where to get help	Frontline staff hear about changes from coworkers, not their manager

When each piece should start

- **Six months out:** name decision makers for each area, set the training plan, schedule backfill, and set shutdown dates for the systems being replaced.
- **90 days out:** start the weekly scorecard, run the first day-in-the-life walkthroughs, and agree on go/no-go criteria.
- **30 days out:** finish training, run final walkthroughs, test devices and logins in every location, and rehearse downtime procedures.
- **One week out:** make the go/no-go call, publish the support schedule, and confirm the command center roster.

Training doesn't stop at go-live, either. KLAS Research's 2026 Clinician EHR Experience report, based on surveys of more than 121,000 clinicians at 92 organizations, points to staggered education: core workflow essentials in week one, with more layered in across the first 90 days. It also found that as little as one hour a year of easy-to-reach ongoing education drives outsized returns (HIT Consultant).

2. Retire "we'll figure it out during go-live"

"We'll figure it out during go-live" is one of the most expensive sentences in an implementation. Sometimes the issue is small. Other times it means a handoff with no owner, a decision waiting on three people, or staff who won't know where to turn when something breaks.

Before cutover, every open item should have an answer to three questions:

1. **Who owns this?** One name, not a team.
2. **What happens if the plan doesn't work?** A fallback, a downtime procedure, or a manual workaround people have actually practiced.
3. **Who needs to know?** The people affected, and how they will hear about it.

A few more things worth settling before day one:

- **A command center with clear roles.** Who triages, who fixes, who decides, and who communicates.
- **One simple way to report problems.** A single number or channel, staffed around the clock for the first days.
- **Escalation rules.** What counts as urgent, who gets called, and how fast.
- **A daily rhythm.** Short huddles with each area at the same time every day, so issues surface early.
- **Go/no-go criteria agreed in advance.** Decide what "ready" means before the pressure is on.

These conversations feel small a month out. On day one, they are the difference between a hard week and a crisis.

Who does what in the command center

Role	What they own
Command center lead	Runs the daily rhythm, owns the issue list, and sets priorities
Triage desk	Logs every issue, assigns a severity, and routes it to the right fixer
Build and technical teams	Fix configuration, interface, and access problems, then confirm the fix with the person who reported it
At-the-elbow support	Helps staff on the floor, spots patterns, and feeds them back to triage
Clinical and operational leads	Make workflow decisions and approve temporary workarounds
Communications lead	Sends one daily update so everyone hears the same thing

Role	What they own
Vendor contacts	Named people, with phone numbers, for problems only the vendor can fix

Agree on severity levels before day one

Everyone should sort issues the same way. A four-level scale works for most organizations. Treat the examples below as a starting point, and set response times with your clinical and operational leaders.

Level	What it means	Example	Response
1: Critical	A patient safety risk, or a whole area can't work	Medication orders aren't reaching the pharmacy	Immediately, all hands
2: High	A key workflow is blocked with no workaround	A clinic can't schedule follow-up visits	Same day
3: Medium	A workaround exists but slows people down	A report prints in the wrong format	Within the week
4: Low	A cosmetic issue or a request for something new	A user wants a field moved	Added to the optimization list

Practice downtime, not just go-live

Downtime procedures often get written for go-live and then forgotten. Run them for real before cutover: which paper or read-only workflows take over, who prints what, and how information gets back into the system when it returns. The same playbook protects you during an outage or a cyberattack long after go-live.

3. The first 90 days decide adoption

Go-live is the start of adoption, not the end of the project. The first 90 days set the habits people keep for years: the workarounds, the shortcuts, and their opinion of the system.

Weeks 1 to 2: stabilize. Keep at-the-elbow support on the floor, especially in high-stakes areas like the emergency department, surgery, and pharmacy. Triage issues daily and close the loop with whoever reported them.

Weeks 3 to 6: fix the friction. Ticket trends show where the build is wrong and where people need more training. Fix the build where it's wrong. Retrain where it isn't.

Weeks 7 to 13: optimize. Move from break-fix to improvement. Replace the workarounds people invented. Tune alerts, order sets, and templates based on how people actually use them.

What helps:

- **Keep super users in the loop.** They are your early warning system. Give them time and a direct line to the build team.
- **Watch the quiet signals.** Overtime, documentation time, and patient throughput often say more than the ticket count.
- **Don't pull support too early.** Release at-the-elbow help when the data says people are ready, not when the original schedule says so.

Technology should make healthcare professionals' jobs easier, not harder. The first 90 days are when you find out whether it does.

What to watch each week

Ticket counts tell you the system is busy. These tell you whether it's working:

- **New tickets by area and type.** A falling total can hide one unit that is still struggling.
- **Repeat issues.** The same question from five people is a training gap or a build problem, not five separate tickets.
- **Time to close high-severity issues.** If it rises after week two, the fix team is overloaded.
- **Documentation and after-hours time.** If after-hours charting keeps climbing, burnout won't be far behind.
- **Throughput.** Discharge times, clinic visits per day, and turnaround for lab and imaging results.
- **Revenue signals.** Late charges, claim edits, and denials. Revenue problems often surface weeks after go-live, once the first claims are paid or denied.

Pay special attention to nursing. In KLAS's 2026 data, 39% of nursing organizations were stuck at the "Struggling" or "Basic" levels of EHR experience, compared with 18% of physician organizations (HIT Consultant).

When it's safe to pull support

Set exit criteria in advance and check them unit by unit, not for the whole organization at once. For example:

- High-severity issues are being closed faster than new ones arrive.
- Ticket volume in the unit has leveled off for two weeks running.
- Super users confirm staff can complete their core workflows without help.
- Throughput and documentation time are back near baseline, or clearly heading there.

Run optimization in short sprints

After stabilization, a sprint format works well. Pick one department, spend a few focused weeks with them, and fix what slows them down most: templates, order sets, alerts, inbox rules, and personal settings. Then move to the next department, and end each sprint with a short report of what changed.

Steady improvement is realistic. Among 51 organizations in the KLAS data that measured their clinicians' experience a second time, the average score rose 6.2 points (HIT Consultant).

4. Finish the job: turn the old systems off

A new platform only pays off fully when the systems it replaced are actually turned off. Until then, you are paying for licenses, servers, support, and security on two sets of tools, and staff are tempted to keep using the old one.

Trinity Health shows what's possible. After moving to Epic, it reported about $80 million in savings from retiring legacy applications, and expects the total to pass $100 million (Chief Healthcare Executive). Its leaders pointed to executive support from the start, monthly reporting to the CEO, a named program, and direct conversations with clinicians who didn't want to give up familiar tools.

What a retirement plan needs:

- **An inventory with owners.** Every legacy app, who uses it, what it costs, and what replaces it.
- **A data plan.** What gets converted, what gets archived for legal and clinical lookback, and how people will find it later.
- **Dates on the calendar.** Shutdown dates set before go-live, not "sometime after things settle down."

- **Contract timing.** Line up shutdowns with renewal and termination windows so the savings start on schedule.
- **A security check.** Every system left running is one more thing to patch and protect.

The best time to plan retirement is before the new system goes live, while the budget and the attention are still there.

A decommissioning playbook

1. **Confirm the replacement covers everything.** Every function the old system performed is handled by the new one, or by a documented workaround, and the business owner signs off.
2. **Freeze it.** Stop new data entry and switch the old system to read-only, with an end date for that period.
3. **Decide what data moves.** Convert what people need for ongoing care and operations. Archive the rest.
4. **Archive for lookback.** Keep historical records searchable in an archive that clinicians, release-of-information staff, and auditors can reach. Medicare's hospital conditions of participation require medical records to be kept for at least five years (42 CFR 482.24), and state laws and other obligations can require longer. Confirm the rules with your legal and health information teams.
5. **Check for legal holds.** Pending lawsuits or audits can require you to preserve data in a specific way.
6. **Cut the connections.** Turn off interfaces, feeds, and reports that touch the old system, and confirm nothing downstream breaks.
7. **Close the contracts.** Give notice on licenses, support, and hosting in line with the termination terms.
8. **Remove access and retire the hardware.** Shut down accounts, servers, and storage, and document how data was disposed of.
9. **Report the savings.** Record what stopped being paid, and when.

What it costs to leave it running

Old systems keep costing money even when nobody logs in: licenses and maintenance, hosting, interfaces, security patching and audits, and the staff who keep them alive. An unpatched system that nobody is watching is also a security risk.

5. Measure value, not activity

Go-live checklists measure activity: tickets closed, people trained, sites live. Value shows up in different numbers, and those numbers need a baseline from before go-live to mean anything.

Area	What to track	When to check
Adoption	Use of key features, workaround rates, how often people fall back to old tools	30 and 90 days
Workflow	Documentation time, turnaround times, steps for common tasks	90 days
Staff	Satisfaction, overtime, and turnover in the most affected roles	90 days and 12 months
Financial	Days in accounts receivable, denial rate, legacy costs retired	6 and 12 months
Patients	Wait times, throughput, portal use	6 and 12 months

Pick a handful, set the baseline now, and report them to the same executive group that approved the investment.

Keep a benefits register

A benefits register is a simple list of the results the investment was supposed to deliver, kept where executives can see it. For each benefit, record:

Field	Example
Benefit	Fewer claim denials caused by missing information
Measure	Initial denial rate, as a share of claims submitted
Baseline	The rate for the six months before go-live
Target and date	The level you expect, and when
Owner	The revenue cycle leader, not the project manager
Data source	The report the number comes from, so it's measured the same way every time

Review the register every quarter for the first year and report it to the group that approved the money. When a benefit is off track, the question isn't who to blame. It's what's in the way: the build, training, workflow, or staffing.

Common pitfalls and how to avoid them

Most of these show up early if someone is looking for them.

Pitfall	Early warning sign	What to do instead
The plan ends at go-live	No budget or staff named for months 4 to 12	Fund optimization and retirement in the original business case
The same experts do everything	A few people are trainers, testers, and super users at once	Map who is doing what, and bring in help where people are doubled up
Training is too early or too generic	Training ends months before go-live, or uses generic scenarios	Train close to go-live with each role's real workflows, and keep teaching after launch
Workarounds become permanent	Staff still keep side spreadsheets or paper logs after week six	Track workarounds as issues, then fix the build or retrain
Support leaves on schedule	Tickets climb again after the support team leaves	Use exit criteria by unit, not calendar dates

Pitfall	Early warning sign	What to do instead
Old systems linger	Renewal notices arrive for systems you replaced	Set shutdown dates before go-live and tie them to contract notice dates
Value is never measured	Nobody can say what improved	Set baselines before go-live and keep a benefits register

What this looks like in practice

An illustrative scenario: a composite built to show how the pieces fit together. It is not a client story, and the details are for illustration only.

A 250-bed community hospital with eight outpatient clinics is moving its hospital and clinics onto one new EHR and retiring two older systems.

- **Six months out,** leaders notice that the same dozen nurses are on the training team, the testing team, and the super user list. They bring in outside trainers for two units and block super user time on the schedule.
- **90 days out,** the weekly scorecard shows night shift training well behind day shift. Sessions move to early morning and evening, with backfill paid.
- **At go-live,** the command center sorts every issue on the same four-level scale. A medication interface problem is caught in the first hours, rated critical, and fixed the same day because the vendor contact was already on the roster.
- **Weeks 3 to 6,** ticket trends show that most clinic questions trace back to one referral workflow. Instead of retraining everyone, the team fixes the build.
- **Weeks 7 to 13,** at-the-elbow support leaves the clinics first, once their exit criteria are met, and stays longer in the emergency department.
- **Months 4 to 12,** the old clinic system shuts off on the date set before go-live, timed to its contract notice window. The benefits register shows which results are on track, and the steering committee funds a second sprint focused on nursing documentation.

Nothing in this scenario is unusual. What makes it work is that each decision was made before the pressure hit.

Questions every executive sponsor should ask

Before go-live

- Does every department have protected training time and backfill?
- In each area, who can make a decision when something stalls?
- Have staff walked through a real day in the new system?
- Are shutdown dates set for the systems this replaces?
- Do we have a baseline for the measures we said would improve?

At go-live

- Does every open issue have one owner and a fallback?
- Do staff know exactly where to report a problem?
- What would make us pause, and who makes that call?

After go-live

- Are we pulling support based on data, or on the original schedule?
- Which workarounds have become habits, and who is fixing them?
- Are the old systems actually off, and are we still paying for them?
- Are we reporting results to the group that approved the money?

Checklist for the project lead

Six months out

- [ ] Name one decision maker for each area
- [ ] Build the training plan and fund backfill
- [ ] Set shutdown dates for every system being replaced
- [ ] Record baselines for the measures the business case promised

90 days out

- [ ] Start the weekly readiness scorecard
- [ ] Run day-in-the-life walkthroughs with real scenarios
- [ ] Agree on go/no-go criteria and severity levels

30 days out

- [ ] Finish role-based training
- [ ] Test devices, printers, and logins in every location
- [ ] Rehearse downtime procedures
- [ ] Publish the command center roster and support schedule

Go-live to day 90

- [ ] Hold daily huddles and close the loop on every issue
- [ ] Track workarounds and repeat issues
- [ ] Pull support unit by unit, using exit criteria
- [ ] Start the first optimization sprint
- [ ] Report the benefits register to the executive sponsor

How Spark helps

Spark works alongside health systems from planning through the months after launch: readiness assessments, go-live command center support, at-the-elbow staffing, optimization, and legacy retirement planning. Our leaders have supported implementations across more than 100 hospitals and 5,000 clinics, and we stay accountable for the outcome, not just the launch date.

Got a go-live on the calendar, or one that didn't deliver what you expected? Start a conversation.

Sources

- Trinity Health talks about saving nearly $80M at ViVE 2026, Chief Healthcare Executive, Feb. 12, 2026
- KLAS Arch Collaborative: Clinician EHR Experience 2026, HIT Consultant, June 26, 2026 (survey of more than 121,000 clinicians at 92 organizations)
- 42 CFR 482.24, Condition of participation: Medical record services, Electronic Code of Federal Regulations
- Spark Healthcare Advisory on LinkedIn, posts on readiness, go-live, and post-launch support, Sept. to Oct. 2026